

# Simulación de protocolos con NS-2

## Práctica 3

### 1. Introducción

NS-2 ha sido el simulador más usado en la investigación sobre redes de comunicaciones. NS-2 es un simulador basado en eventos discretos, escrito en C++, que cubre un vasto número de aplicaciones, de protocolos, de tipos de red, de modelos de tráfico, entre otros. NS-2 puede descargarse en <http://www.isi.edu/nsnam/ns/>.

NS-2 viene acompañado de otras herramientas. NAM (Network Animator) para animar las simulaciones escritas para NS, (Xgraph) para trazar los resultados de simulaciones, GT-ITM para generar topologías de red con un gran número de nodos, entre otras.

Por otro lado, el protocolo TCP es el protocolo que transporta más del 90 % del tráfico en Internet. TCP tiene como responsabilidad principal adaptar la velocidad de transmisión al ancho de banda disponible en la red, reducir la congestión y asegurar una transmisión confiable y ordenada de extremo a extremo. Comprender el funcionamiento de TCP es primordial para dimensionar adecuadamente Internet.

### 2. Objetivos

Esta práctica tiene como objetivo primordial conocer los fundamentos de uso del simulador NS-2. Para ello, se escribirán y ejecutarán tres escenarios simples de simulación y se explicarán los eventos ocurridos, ayudándose con NAM.

Existen al menos tres formas de instalar NS-2. La primera, consiste en descargarla a partir de repositorios y simplemente ejecutar el simulador. La segunda, se conoce como *all in one*, que consiste en descargar un archivo comprimido que incluye todos los componentes del simulador, además de un script que sirve para compilar e instalar el simulador en el sistema. La tercera, que será por la que optaremos en esta práctica, se conoce como *piece by piece*. En esta, se descargan uno a uno los distintos componentes del simulador, se compilan en orden y al final se obtiene el simulador funcional.

La principal razón de elegir la tercera opción es que, además de conocer los fundamentos del simulador, esta práctica también le servirá como experiencia para saber cómo compilar este tipo de sistemas. Además de NS-2, una gran parte de software libre se compila exactamente de la misma forma. Las dos primeras opciones, no permitirían adquirir esta experiencia.

### 3. Procedimiento

En esta práctica, usted transcribirá los scripts señalados más abajo. Se trata sobre todo, de observar cómo se configuran escenarios básicos en NS-2, desde el más simple hasta uno más elaborado. Observe también en qué partes de los scripts se configuran parámetros de las distintas capas del Modelo TCP/IP. Algunas capas son más detalladas que otras, así que ese será otro aspecto que podrá observar.

Aunque no es indispensable para realizar la práctica, si usted está interesado en consultar alguna referencia, se sugieren las de Altman & Jiménez [1] y la de Teerawat Issariyakul [2].

Para ejecutar los scripts de abajo, siga las instrucciones de compilación de NS-2 *piece by piece*. Entonces, agregue a la variable de ambiente PATH la ubicación donde se encuentra el ejecutable del simulador. Dadas las instrucciones, dicha ubicación sería en la carpeta bin de su directorio HOME.

### 4. Desarrollo

Escriba los siguientes tres scripts, ejecútelos en NS-2 y analice su utilidad. Los scripts se escriben en el lenguaje OTCL, el cual es interpretado. Si cuenta con una instalación funcional de NS-2, la ejecución de un script NS-2 se realiza en una terminal de comandos como sigue:

```
$ ns script1.tcl
```

suponiendo que haya guardado su script en el archivo “script1.tcl”. Para la entrega de esta práctica, se le solicita mostrar cómo compila NS-2 para luego ejecutarlo. Por otro lado, para cada script se le harán algunas preguntas con respecto de su funcionalidad.

Finalmente, se recomienda ampliamente que no copie y pegue los scripts a partir de este archivo. En su lugar, escríbalos usted mismo. Las razones de esta sugerencia son varias; primeramente, escribir usted mismo los scripts le permite analizar cada línea; segundo, al copiar y pegar texto de un archivo PDF se obtienen con frecuencia caracteres no soportados por un editor de texto, o peor aún, invisibles, lo que complicaría significativamente su ejecución. La última recomendación es que usted mismo escriba sus scripts, no los pida a alguno de sus compañeros.

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf
#Define a 'finish' procedure
```

```
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}
#Call the finish procedure after
#5 seconds simulation time
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

Realice ahora el mismo ejercicio con el siguiente ejemplo:

```
#Create a simulator object
set ns [new Simulator]
#Open the nam trace file
set nf [open out.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace
    #Close the trace file
    close $nf
    #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

#Create two nodes
set n0 [$ns node]
set n1 [$ns node]

#Create a duplex link between the nodes
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

```
#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0

# Create a Null agent (a traffic sink) and
# attach it to node n1
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0

#Connect the traffic source with the traffic sink
$ns connect $udp0 $null0

#Schedule events for the CBR agent
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"

# Call the finish procedure after 5 seconds of
# simulation time
$ns at 5.0 "finish"

#Run the simulation
$ns run
```

Finalmente, ejecute el siguiente script:

```
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the trace files
set file1 [open out.tr w]
```

```
set winfile [open WinFile w]
$ns trace-all $file1

#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2

#Define a 'finish' procedure
proc finish {} {
    global ns file1 file2
    $ns flush-trace
    close $file1
    close $file2
    exec nam out.nam &
    exit 0
}

#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mb 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mb 30ms DropTail

#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient left
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
#Set Queue Size of link (n2-n3) to 10
```

```
$ns queue-limit $n2 $n3 20

#Set up a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552

#Setup an FTP application over the TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

#Setup a UDP flow
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2

#Setup a CBR traffic source over the UDP flow
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
$cbr set random_ false
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"

# next procedure gets two arguments: the name of the
# tcp source node ("tcp") and the name of output file.
proc plotWindow {tcpSource file} {
    global ns
```

```
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
set wnd [$tcpSource set window_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}

$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 125.0 "finish"
$ns run
```

## 5. Cuestionario

1. Describa qué hace cada script ejecutado en la práctica. Analice el código y explique en qué parte se configura cada capa del Modelo TCP/IP.
2. ¿Cómo podría usarse NS-2 para explicar la relación entre el retardo de propagación y el de transmisión en un enlace? ¿Qué parte de un script TCL debería modificar para ejemplificar esto?
3. Para el segundo ejemplo de esta práctica, modifique el código para que el tamaño del paquete sea igual a 1,500 bytes. ¿Qué diferencias observa en la animación con NAM?
4. Investigue la disponibilidad de simuladores de red y realice una tabla que compare sus características.
5. Describa cómo se configura cada una de las capas del Modelo TCP/IP en los scripts 2 y 3.

## Referencias

- [1] Eitan Altman and Tania Jiménez. NS simulator for beginners. *Synthesis Lectures on Communication Networks*, 5(1):1–184, January 2012.
- [2] Teerawat Issariyakul and Ekram Hossain. *Introduction to Network Simulator NS2*. Springer US, 2012.